

## JOB SHOP SCHEDULING WITH ALTERNATIVE MACHINES USING GENETIC ALGORITHMS

ZALINDA OTHMAN<sup>1</sup>, KHAIRANUM SUBARI<sup>2</sup> AND NORHASHIMAH MORAD<sup>3</sup>

**Abstract.** Recently, an integration of manufacturing functions has gained interest from a number of researchers, particularly in production planning and scheduling. These two functions play important roles in production, especially to ensure the availability of manufacturing resources needed to accomplish production tasks. This paper explores the use of Genetic Algorithms (GA) in solving the problem associated with the integrated production scheduling. The integrated problem considers the alternative routing for operations of each job during the creation of schedules. In alternative routing there is a choice of machines on which to perform the operations. These machines take different amount of time to process the same operation. By considering the alternative routing, the possible solutions for the scheduling problem become very vast. As a robust approach, GA is used to find the most promising solution. The optimization of this problem involves several objectives, namely minimizing makespan, minimizing processing cost, and minimizing number of rejects. It also takes into account the constraints on operations sequence. The proposed solution was compared with previous approaches, and the numerical simulations showed promising results.

**Keywords:** Genetic algorithms, job shop scheduling, alternative machines

**Abstrak.** Sejak kebelakangan ini integrasi fungsi-fungsi pembuatan telah menarik minat sejumlah penyelidik terutamanya dalam perancangan pengeluaran dan penjadualan. Kedua fungsi ini memainkan peranan penting di dalam proses pengeluaran, terutamanya bagi memastikan sumber-sumber pembuatan yang diperlukan untuk melaksanakan proses pengeluaran telah tersedia. Kajian ini telah melihat integrasi penjadualan pengeluaran dengan objektif utamanya ialah menilai keupayaan Algoritma Genetik (GA) dalam menyelesaikan masalah tersebut. Masalah integrasi ini menimbangkan penghalaan alternatif untuk operasi-operasi bagi setiap tugas semasa pembentukan jadual-jadual. Dalam penghalaan alternatif terdapat pilihan ke atas mesin-mesin yang akan memproses suatu operasi. Mesin-mesin ini mungkin mengambil masa yang berbeza untuk memproses suatu operasi yang sama. Dengan mengambilkira penghalaan alternatif, penyelesaian yang mungkin untuk masalah penjadualan menjadi terlalu besar. Pendekatan GA digunakan bagi mencari penyelesaian terbaik. Pengoptimuman masalah ini melibatkan beberapa objektif, iaitu *makespan*, kos pemprosesan dan bilangan yang ditolak. Kami telah membandingkan pendekatan yang dicadangkan dengan beberapa pendekatan daripada penyelidik terdahulu dan hasil simulasi telah menunjukkan keputusan yang memuaskan.

**Kata kunci:** Algoritma genetik, penjadualan bengkel kerja, mesin alternatif

<sup>1&2</sup> School of Mechanical Engineering, Engineering Campus, University Science Malaysia, Seri Ampangan, 14300 Nibong Tebal, Seberang Prai Selatan, Pulau Pinang.

<sup>3</sup> School of Industrial Technology, University Science Malaysia, 11800, Minden, Pulau Pinang, Malaysia.  
Email: zalinda@eng.usm.my, meanum@eng.usm.my and nhashima@usm.my

## 1.0 INTRODUCTION

Many studies on scheduling problems have been done by researchers since 1950 [1]. After decades, scheduling problem still attracts the interest of researchers due to its characteristic as a combinatorial problem. Generally, scheduling can be described as the allocation of a set of resources in a period of time, to perform a set of jobs. The scheduling problems depend on the shop floor environment, such as flow shop, job shop, and open shop. For each environment, a different scheduling approach is needed [2]. In this paper, the scheduling problem is considered as a static job shop problem, taking into account the available alternative machines. By allowing a choice of operation sequences and/or machines at the scheduling stage, a more flexible and effective schedules can be produced. Since part routing is a process planning activity, choosing alternative machines during scheduling is in fact integrating the process planning and the scheduling tasks. This scenario can be referred to as integrated process planning and scheduling or, for brevity, integrated scheduling [3].

There are various job shop scheduling approaches reported in the literature such as heuristic or dispatching rules, mathematical models, neural network, simulated annealing, tabu search, fuzzy logic, and genetic algorithm (GA) [4]. In this paper, the GA approach is used because it is one of the approximate approaches that can handle a large searching space, as in this integrated scheduling problem. There is no complex mathematical formula in developing GA and it can also be understood easily. Furthermore, GA can produce a set of acceptable solutions, compared to other search methods such as simulated annealing, which gives only one solution. These acceptable solutions will give a few choices to the person dealing with the shop floor production.

## 2.0 INTEGRATED SCHEDULING

Traditionally, scheduling function receives input from a process plan that specifies a unique choice of machines for each operation, and a unique sequence of operations. In integrated scheduling, the input comes from a more flexible process plan. The flexible plan allows a choice of operation sequences and/or machines [3]. Therefore, by allowing choices of operation sequences and/or machines, there are two situations that should be considered which relate to the processing time, and the order of the operation. These machines may be the same and have the same processing time for a given operation, or the alternative machine may have different processing time. The routes of the process and the order of the operations may change depending on the choice of the machines.

Therefore, two standard assumptions listed in [5] could be relaxed in the integrated problem. Those assumptions are:

- (1) *J1*: All processing times are fixed and sequence-independent

(2)  $J2$ : The processing order per job is known and fixed

The integrated scheduling problem can be regarded as:  $n|m|G, \{J1, J2\} | f$ , where  $n$  is a number of job,  $m$  is a number of machine,  $G$  is for general job shop, and  $f$  is a performance measurement [3]. In other words, the integrated scheduling problem can be described as follows: there are a set of machine and a set of jobs which consist of chain of operations, each of which needs to be processed. Each machine can process at most one operation at the time, and each operation can be processed by more than one machine with different fixed processing time, and cost [6].

Alternative routings (which involves alternative machines) are one of the scheduling issues in a real production environment. The theory and practical aspect of scheduling problems are discussed in [7]. These alternative machines may have the same processing time or may be different. Besides that, scheduling in a real working life faces multiple objectives. Therefore, the integrated problem can be seen as one way of closing the gap between theoretical and practical scheduling.

There is an increase number of work done on this subject and various approaches have been used to solve the problem. Sundaram and Fu [8] were among the earlier researchers who showed that there is a need for integrating the process planning and scheduling functions in manufacturing for productivity improvements. They used heuristic procedures for solving the integrated problem. Other approaches are integer programming [9], rule-based approach [10], simulated annealing [3], and GA approach [6,11-13].

Most of the integrated approaches, as mentioned above, have concentrated on the time aspects of the alternative machines such as makespan and average flowtime. Other evaluation criteria, such as total process plan cost, total tardiness, number of late orders etc., which are also of practical interest and importance have not been very well studied [14]. As mentioned by Tan and Khoshnevis [14], the optimisation aspects cover more than just the amount of time that it takes to process the jobs. The study by Morad and Zalzal [13], includes other criteria as the performance measurements. These include the total number of rejects and total processing costs. However, the schedule builder was not flexible to handle a big problem. In this study, the authors improved this by developing a more flexible schedule builder and tested it by using a different set of data.

### 3.0 GENETIC ALGORITHM FORMULATION

In this work, each chromosome is represented by a list of job order, respective operation and machine sequence [15,16]. An example of the chromosome representation is shown in Figure 1.

In Figure 1, the chromosome consists of two jobs with two operations each. The order of the job to be done is the first job, Job1, followed by Job3. In Job1, the first operation (i.e. Op1) is to be done at machine Mc1, then followed by operation two (i.e. Op2) at machine Mc3. Similarly, for job number 3 (i.e. Job3), the first operation is to be done at Mc2 and the second operation is to be done at Mc1.

Job1	Op1	Mc1	Op2	Mc3	Job3	Op1	Mc2	Op2	Mc1
------	-----	-----	-----	-----	------	-----	-----	-----	-----

**Figure 1** An example of the chromosome used: Op for operation; Mc for machine

In this problem, the schedule builder consists of two cross-over operators. They are order-based operator and plan-resource operator.

(1) Order-based operators

Order-based crossover [17] changes the order of jobs; the order of jobs in the selected position in one parent is imposed on the corresponding tasks in the other parent. The order-based mutation interchanges the positions of the jobs at random.

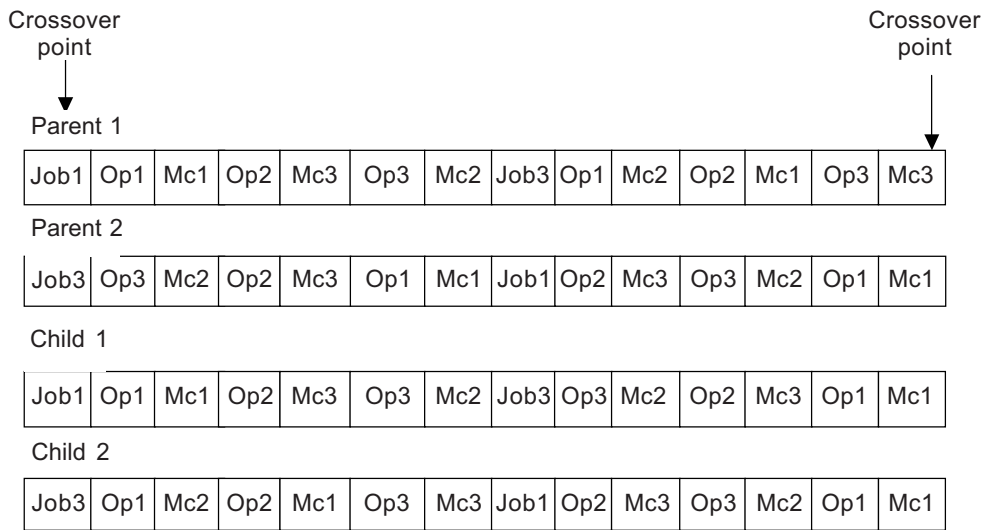
(2) Plan-resource operators

Plan-resource crossover, as developed by Morad [15], changes the process plan as well as the machine to perform operation. The sequence of the job remains the same. Meanwhile, in the mutation, the process plans as well as the machines are re-chosen at random. Figure 2 shows an example of plan-resource crossover.

In Figure 2, the two parents (Parent 1 and Parent 2) will be swapping information to produce new offsprings (i.e. Child 1 and Child 2). Child 1 is created by combining information of job 1 from Parent 1, and job 3 from Parent 2. Meanwhile, Child 2 takes the information of job 1 from Parent 2, and job 3 from Parent 1.

To ensure the effectiveness of the GA, the search space is reduced within active schedules only. In active schedule, the processing sequence is such that no operation can be started any earlier without delaying some other operations or breaking the technological constraints. It is known that the optimal solution to a job shop scheduling problem resides within the set of active schedules, which is much smaller than the set of semi-active. In order to search within the active schedules, the GA incorporates the well-known Giffler & Thompson (GT) [18] algorithm as a schedule builder.

The GT algorithm is an approach of generating schedules that have good makespan. However, the schedule builder developed in this work is slightly different from the GT algorithm. The difference is the way the operation is chosen at a specific machine.

**Figure 2** Plan-resource crossover

The selection criteria of the proposed schedule builder is not only based on the minimum finishing time at the machine, but it is also based on the job sequence that is determined by the chromosomes. In this case, the best order of jobs is also deemed important in scheduling. The developed GA will find the best order of jobs, along with the operations, and machines to process the parts. To be more flexible, other performance measurements are included in this schedule builder such as the number of rejects and the processing cost.

In this study, the GA parameters with respective values are shown in Table 1. These values are chosen based on the range used by previous researchers [19, 20].

**Table 1** Values of the GA parameters

Factor	Parameter value
Population size, NIND	140
Probability of order-based crossover, XOVr	0.5
Probability of order-based mutation, Pm	0.001
Probability of plan-resource crossover, XPRX	0.5
Probability of plan-resource mutation, MUPRX	0.01

#### 4.0 INTEGRATED SCHEDULING SIMULATION RESULTS AND DISCUSSION

In this study, a number of experiments have been conducted in order to asses the GA-based schedule builder. Table 2 summarizes the experiments conducted in this work.

**Table 2** List of experiments and the objectives

Experiment	Objective of the experiment
Experiment 1	To show how the schedule builder can be used to minimize different objective functions.
Experiment 2	To compare how the performance of the schedule builder is better than previous work with different precedence constraints.
Experiment 3	To compare the performance of the schedule builder with various approaches from previous researchers.

##### 4.1 Experiment 1

In this experiment, three objectives were used for performance measurement. These objectives are:

*F1* – minimizing total completion time of all jobs or the makespan.

*F2* – minimizing total number of rejects, i.e. the total number of rejects can be calculated by adding all the numbers of scraps produced. This is illustrated below;

$$F2 = \sum_{l=1}^n Y_l^s = \sum_{l=1}^n Y_l^o \left( \sum_{j=2}^{op} k_j^s \right) \quad (1)$$

*F3* – minimizing total processing cost, i.e. the total processing cost can be calculated by summing all the processing costs for all operations as shown below;

$$F3 = \sum_{l=1}^n N_l^i \sum_{j=1}^{op} \left[ k_{lj}^i X_{lj}^i - k_{lj}^s X_{lj}^s + k_{lj}^i f(Y_{lj}^i) \right] \quad (2)$$

Equations (1) and (2) are taken from Singh [21] and the notations used are described below;

$Y_l^s$  is the scrap unit;

$Y_i^o$  is the output unit;

$k_j^i, k_j^s$  are the technological coefficients per unit output for input and scrap respectively;

$X_j^i, X_j^s$  are the unit average cost of input and scrap respectively;

$f(Y_j^i)$  is the processing cost per unit;

$n$  is the total number of jobs;

$l$  is the job number;

$op$  is the total number of operations;

$N^i$  is the number of input units.

In the experiment, for each run, selected objective function ( $F1$ ,  $F2$  or  $F3$ ) is being minimized individually. The results are shown in Table 3. Experiment 1(a), 1(b), and 1(c) correspond to minimize makespan only, minimize total processing cost only and minimize total number of reject only, respectively.

**Table 3** Minimizing objective functions individually for fixed operation sequence

Experiment	Makespan	Number of rejects	Total processing cost
1(a)	918	13	721
1(b)	1123	15	515
1(c)	1534	10	829

In this experiment, it is shown that the best acceptable value for each objective is 918 unit time for makespan, 515 unit dollar for processing cost, and 10 unit items for the number of rejects.

## 4.2 Experiment 2

In order to show the performance of the schedule builder, the developed GA is compared to the previous works with different precedence constraints. First, a simulation has been conducted which minimize makespan for random operations sequence. This is the example of a situation with no precedence constraint. In this case, the schedule builder will take the makespan as the objective function, the other two variables are calculated based on the final schedule. Table 4 shows the results for the developed GA with Morad's results.

**Table 4** A comparison with minimizing makespan only for random operation sequence

	The proposed GA	Morad
Makespan (unit time)	971	1265
Total processing cost (unit dollar)	695	1268
Number of rejects (unit)	14	16

From Table 4, the results show that the overall results for three objectives have improved as compared to Morad's. The best acceptable value for makespan is 971 unit time, compared to 1265 unit time; for total processing cost is 695 unit dollar compared to 1268 unit dollar; and for the number of rejects is 14 unit items compared to 16 unit item. Hence the schedule builder increases the overall performance of the proposed GA.

Secondly, a further comparison has been conducted on a real industrial problem. In this case, there is a strict precedence constraint imposed on the operations sequence. Other researchers, using other approaches, had also attempted to solve this problem. Palmer [3] used simulated annealing (SA), and has reimplemented the Khoshnevis and Chen's [22] dispatching rules algorithms on the problem set. Morad attempted to solve the problem by using the GA technique. In this simulation, the objective function is mean flowtime + 2x total tardiness ( $MFT2$ ), which is shown as follows:

$$MFT2 : \sum F_j / n + 2 \sum T_j \quad (3)$$

While there are  $n$  jobs  $J_j$  ( $j = 1 \dots n$ ), the flowtime of a job ( $F_j$ ) is the difference between its completion time and its due date. A job's tardiness ( $T_j$ ) is its positive lateness. Therefore, the mean flowtime is the average completion time of all jobs and total tardiness is the sum of all jobs' overrun. The results from various approaches are shown in Table 5.

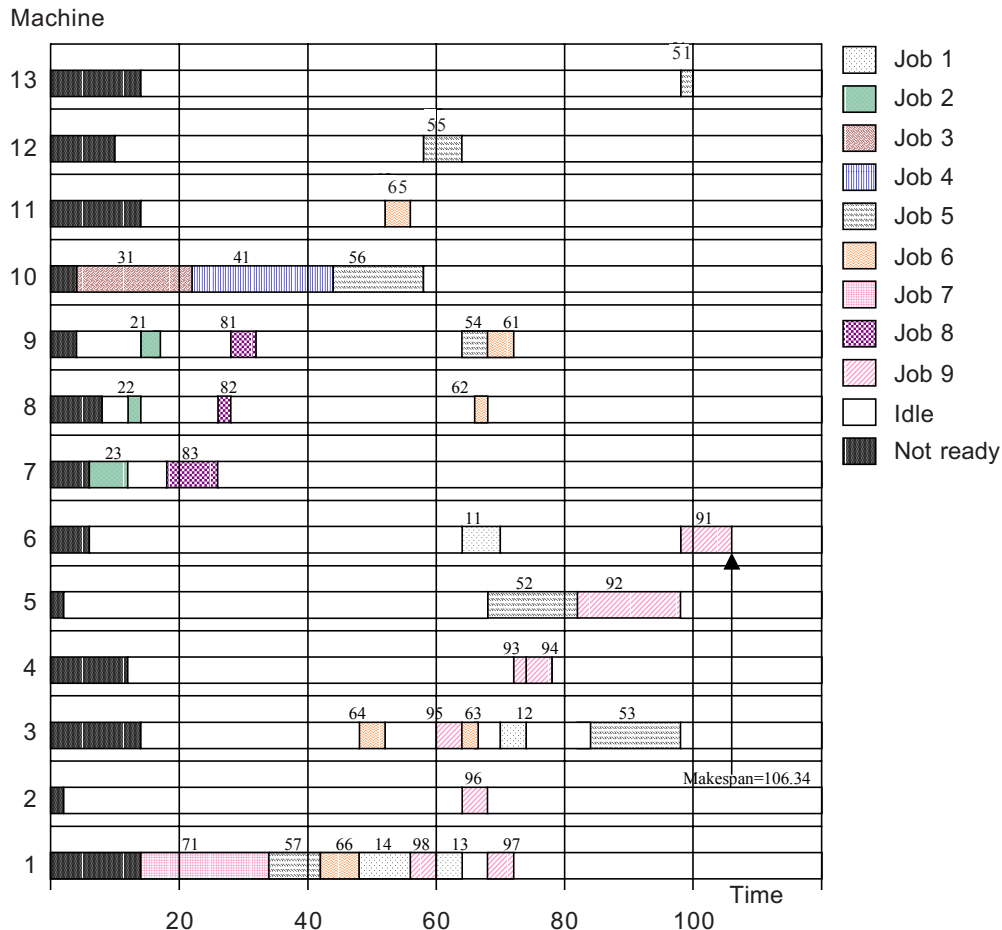
Table 5 shows that the proposed GA is able to provide better makespan compared to other approaches. The makespan is reduced by 1.8%. However, the mean flowtime is slightly higher than the GA developed by Morad. The Gantt Chart shown

**Table 5** Results obtained from various approaches on industrially based problem set

Approach	Objective function	Mean flowtime	Makespan
The proposed GA	MFT2	45.57	106.34
GA (Morad)	MFT2	43.22	110.75
SA (Palmer)	MFT2	50.5	108.32
Heuristic (K&C)	MFT2	63.37	145.47



in Figure 3 is a schedule obtained for the above makespan value. The schedule shows the sequence of each operation for each job. There are 9 jobs and 13 machines. In this case, the sequence of operations for each job is fixed. It can be seen that the total jobs completed when the final job 9 is finished at 106.34 unit time.



**Figure 3** A Gantt chart for the strict precedence constraint situation

### 4.3 Experiment 3

In this experiment, the performance of the proposed GA is being compared with the data sets taken from Nasr and Elsayed [9], and Dutta [23]. The comparisons are made with the mixed-integer programming by Nasr and Elsayed, and artificial intelligence by Dutta. Two different performance measurements are used to compare both approaches. Mean flow time is used to compare the proposed GA with Nasr and Elsayed's method, and makespan is used to compare the proposed GA with Dutta's method. The result of the comparison is shown in Table 6.

From Table 6, it is shown that the proposed GA produced shorter mean flow and makespan, compared to other approaches. In comparison with Nasr and Elsayed, the mean flow is reduced by 4%. The same result is given by [6]. One of the possibility is due to the small size of the data set and this value is the best solution for this case. In another comparison, the proposed GA produced the shorter makespan by 9.5%.

**Table 6** The performance of the proposed approach compared to other approaches

Approach	Performance measurement	
	Mean flow time	Makespan
Proposed GA	11.75	76.0
Nasr and Elsayed	12.25	–
Dutta	–	84.0

## 5.0 CONCLUSION

This paper has looked into two aspects of flexibility, namely sequence flexibility and machine tools flexibility in scheduling. The flexibility in sequence involves two situations. Those situations are fixed operations sequence and random operations sequence. Meanwhile, the flexibility in machine tools involves machine alternatives. These machines are able to carry out certain operations only. This integrated scheduling can be implemented in a system, which needs a higher degree of flexibility, such as flexible manufacturing system (FMS). There are many approaches used to handle this problem. There are approaches such as heuristic, simulated annealing, and artificial intelligence. In this paper, one of the search methods, the GA is used to solve the integrated scheduling problem. Based on the simulations conducted in this work, the GA-based schedule builder is able to solve the given problems and has shown some promising results. The flexibility of the proposed approach can be observed by the ability of the schedule builder in solving the integrated scheduling with different situations, such as the different number of machines, the different number of jobs, the precedence constraints, and the different performance measurements. In experiment 1, the proposed GA is able to solve problem with different objective functions. In this paper, the GA is used to solve a scheduling problem that considers not only the time aspect of the production, but also the manufacturing aspects such as the number of rejects and the total processing costs. While in experiment 2, the GA had solved two different situations, one with strict precedence constraint, and the other without precedence constraint. These two aspects also have significant impact in scheduling. Besides testing the algorithm on simulated data, the proposed approach is also used to solve industry based problem set that involves 9 jobs and 13 machines. Lastly, experiment 3 had shown the ability of the GA to solve integrated problem, compared

with other approaches from previous work. The scheduling problem discussed in this paper is a static job shop problem. There is no disturbance on the shop floor such as machine breakdown, rush job, and so on. It is also a deterministic problem as the variables such as processing time and number of jobs, are known. In the future, a further study on GA is necessary in order to investigate the ability of GA in solving a stochastic and dynamic scheduling problem, such as the possibility of having the dynamic elements like the machine breakdown, the reliability of the machine, the continuous job arrival, and different processing time. In this paper, the comparison is made based on the quality of the solution only. In this work, the time taken for GA to converge to the acceptable results is not being considered. It is thought that the time can be reduced by other methods such as improving the programming and also implementing the parallelism aspect of the GA and the hardware.

## REFERENCES

- [1] Johnson, S. M. 1994. Optimal Two- and Three- Stage Production Schedules with Setup Times Included. *Naval Research Logistics Quarterly*. 1: 61-67.
- [2] Melnyk, Steven A., S. K. Vickery, and P.L. Carter. 1986. Scheduling, Sequencing and Dispatching: Alternative Perspectives. *Production and Inventory Management* 2<sup>nd</sup> quarter: 58-67.
- [3] Palmer, J. 1996. A Simulated Annealing Approach to Integrated Production Scheduling. *Journal of Intelligent Manufacturing*. 7: 163-176.
- [4] Subramaniam, V., T. Ramesh, and A. S. Raheja. 2002. Exploiting the Flexibility of Multiple Job Routes in a Dynamic Job Shop. *Studies in Informatics and Control: with Emphasis on Useful Applications of Advanced Technology*. 11(2). [http://www.ici.ro/ici/revista/sic2002\\_2](http://www.ici.ro/ici/revista/sic2002_2)
- [5] Rinnooy Kan, A.H.G. 1976. *Machine Scheduling Problems: Classification, Complexity and Computations*. Martinus Nijhoff.
- [6] Ilkyeong, M., and J. Lee. 2000. *Genetic Algorithm Application to the Job Shop Scheduling Problem with Alternative Routings*. BK21 Logistics Team. Industrial Engineering Pusan National University.
- [7] Pinedo, M. 1995. *Scheduling: Theory, Algorithms, and Systems*. New Jersey: Prentice Hall.
- [8] Sundaram, R.M., and S. Fu. 1988. Process Planning and Scheduling. *Computers and Industrial Engineering*. 15(1/4): 296-301.
- [9] Nasr, N., and E. A. Elsayed. 1990. Job Shop Scheduling with Alternative Machines. *International Journal of Production Research*. 28(9): 1595-1609.
- [10] Ali, R. B., and B. Koshnevis. 1996. Integration of Machine Requirements Planning and Aggregate Production Planning. *Production Planning & Control*. 7(3): 292-298.
- [11] Jain, A. K., and H. A. Elmaraghy. 1997. Production Scheduling/Rescheduling. *International Journal of Production Research*. 35(1): 281-309.
- [12] Husbands, P., and F. Mill. 1991. *Simulated Co-Evolution as the Mechanism for Emergent Planning and Scheduling*. San Mateo, California: Morgan Kaufmann. 264-269.
- [13] Morad, N., and A. M. S. Zalzala. 1999. Genetic Algorithms in Integrated Process Planning and Scheduling. *Journal of Intelligent Manufacturing*. 10: 169-179.
- [14] Tan, W., and B. Khoshnevis. 2000. Integration of Process Planning and Scheduling: A Review. *Journal of Intelligent Manufacturing*. 11(1): 51-63.
- [15] Morad, N. 1997. Optimisation of Cellular Manufacturing Systems using Genetic Algorithms. PhD Thesis: University of Sheffield.
- [16] Bruns, R. 1993. *Direct Chromosome Representation and Advanced Genetic Operators for Production Scheduling*. San Mateo, California: Morgan Kaufmann. 352-359.
- [17] Syswerda, G. 1991. *Schedule Optimization using Genetic Algorithm*. Handbook of Genetic Algorithms (Lawrence Davis ed.): 332-349.

- [18] Giffler, B. and G. T. Thompson. 1960. Algorithm for Solving Production Scheduling Problems. *Operations Research*. 8: 487-503.
- [19] Kumar, H. N. S. and G. Srinivas. 1996. A Genetic Algorithm for Job Shop Scheduling: A Case Study. *Computers in Industry*. 31: 155-160.
- [20] Drake, P. R. and I. A. Choudhry. 1997. From Apes to Schedules. *Manufacturing Engineer*. February: 43-45.
- [21] Singh, N. 1996. *Systems Approach to Computer Integrated Design and Manufacturing*. New York: John Wiley.
- [22] Khoshnevis, B. and Q. Chen. 1990. Integration of Process Planning and Scheduling Functions. *Journal of Intelligent Manufacturing*. 1: 165-179.
- [23] Dutta, A. 1990. Reacting to Scheduling Exceptions in FMS environments. *IIE Transactions*. 22(4): 300-314.